

Inpainting by Joint Optimization of Linear Combinations of Exemplars

Brendt Wohlberg

EDICS: IMD-ANAL

Abstract—Exemplar-based methods, in which actual image blocks are used to fill in missing content, have achieved state of the art performance in image inpainting. The majority of these adopt a progressive approach, filling in the missing region inwards from the boundary. The final result is highly dependent on fill order, and while significant progress has been made on the choice of this order, the greedy nature of such a process leads to artifacts in some cases. The alternative exemplar-based approach proposed here is defined via joint optimization of a single functional, simultaneously assigning an estimated value to the entire inpainting region. The results are found to be highly competitive with other recent inpainting methods.

Index Terms—image inpainting, image completion, exemplar, patch, block, sparse representation

I. INTRODUCTION

Image inpainting [1] is an image restoration problem with the goal of replacing a specified contiguous image region with visually “reasonable” or “plausible” content so that the inpainted region is not clearly noticeable to a human viewer. Exemplar-based methods, which have been successful in problems such as denoising [2], [3] and superresolution [4], [5], have also been found to give very good results for texture synthesis [6] and inpainting [7], [8], [9], [10], [11]. The common theme of these methods is the use of a set of actual image blocks, extracted either from the image being restored, or from a separate training set, as an image model.

The usual approach to exemplar-based inpainting is to progressively fill in blocks on the boundary of the inpainting region using matching blocks in the known region of the same image [7], [9], [10], [11]. The fill order is chosen to minimize artifacts by giving priority to blocks containing significant edges leading into the inpainting region. This progressive approach constitutes a greedy optimization, and while recent fill order criteria [11] give significantly improved results over the original proposal [7], any greedy approach will be unable to avoid suboptimal choices in certain configurations. The approach proposed here, in contrast, is a global optimization, constructing the inpainting solution as the minimum of a single functional defined in terms of linear combinations of image blocks. While the method of Wexler *et al.* [8] may also be considered as a global optimization, it is defined and computed in an entirely different way, and it gives significantly different behavior in difficult inpainting examples.

The author is with Los Alamos National Laboratory, Los Alamos, NM 87545, USA. Email: brendt@lanl.gov, Tel: +1 505 667 6886, Fax: +1 505 665 5757. This research was supported by the NNSA’s Laboratory Directed Research and Development Program.

II. JOINT OPTIMIZATION OF EXEMPLAR DICTIONARIES

The image model of the proposed approach is that each block of the inpainted region should be a sparse linear combination of other image blocks, agreeing with known image content (i.e. external to the inpainting region) where the block intersects the known region of the image, and agreeing with all other image blocks with which it overlaps. The blocks used to construct the dictionaries for these linear combinations are extracted from the known region of the image being restored (but could also be extracted from a separate training image set), and are chosen depending on the fraction of known content in each block to be inpainted. When this fraction exceeds a predetermined threshold, the entire image is searched for blocks that match on the known part of the target block. Source blocks for blocks with insufficient known content, in contrast, are collected from the part of the image spatially adjoining the inpainting region. This model is applied to multi-band (e.g. RGB color) images by taking the vector representation of an image block as the concatenation of all image bands on the block support.

This model may be expressed as the minimizer of a global functional that

- 1) penalizes the mismatch between solution blocks and known pixels,
- 2) penalizes the mismatch between overlapping parts of different solution blocks, and
- 3) penalizes the ℓ^1 norm of the linear combination coefficients to encourage a sparse, low complexity, solution.

The overlapping of image blocks is critical, as it enables propagation of information from the exterior of the inpainting region to blocks entirely within the interior, as illustrated in Fig. 1, and also reduces blocking artifacts.

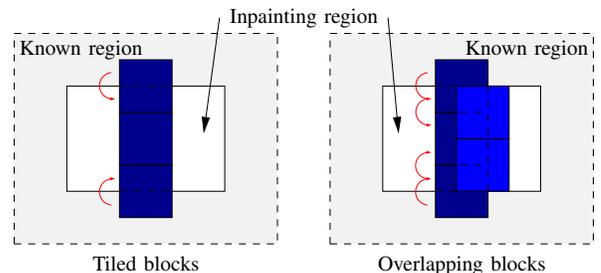


Fig. 1. A penalty on the mismatch with known image pixels can be applied to image blocks that cross the boundary of the inpainting region, but not to blocks interior to the inpainting region. In an overlapping block structure, an additional penalty on the mismatch between overlapping blocks allows the mismatch penalty on the known pixels to propagate to interior blocks.

To reduce the complexity of computing the mismatch between overlapping parts of different blocks, the blocks are arranged in N_G indexed grids, with the overlap being produced by an offset of the entire grid, as indicated in Fig. 2. This structure allows the total block overlap mismatch to be computed as the mismatch between the grids, without having to track overlapping parts of individual blocks. Each block is indexed by a grid and block number, block k, l being the l^{th} block in the k^{th} grid, and $\Phi_{k,l}$ and $\alpha_{k,l}$ are the dictionary and coefficients respectively for block k, l . The image to be inpainted is denoted by vector \mathbf{s} .

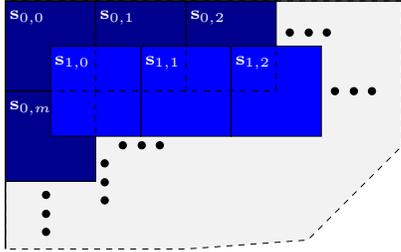


Fig. 2. Structure of overlapping block grids.

The following linear operators are defined:

- $B_{k,l}$ “Extract” block k, l from an image.
- $B_{k,l}^T$ “Insert” block k, l into a zero-valued image.
- $R_{k,l}$ “Extract” the known part of block k, l . The results is a vector with length corresponding to the number of known pixels in the block.
- $R_{k,l}^T$ “Insert” the known part of block k, l into a zero-valued block.
- $Q_{k,l}$ “Extract” block k, l from a vector representing the unknown region of image, inserting zeros where the block does not intersect inpainting region. The result is a zero-valued vector if block k, l has no intersection with the inpainting region.
- $Q_{k,l}^T$ “Insert” block k, l into zero-valued vector representing the inpainting region. The result is a zero-valued vector if block k, l has no intersection with the inpainting region.
- G_k Apply a mask to the inpainting region vector, zeroing out any pixels that are not in grid k .
 $G_k = \text{diag} \left(\sum_l Q_{k,l}^T Q_{k,l} (1 \ 1 \dots)^T \right)$.

Note that in this formulation, operator $R_{k,l}$ maps block k, l to a vector that excludes any unknown (to be inpainted) pixels in the block. This is computationally more efficient, and also leads to a more well-conditioned linear system, than the original approach to this inpainting method [12] in which the corresponding operator zeroed these unknown pixels but retained the original block dimensionality.

Individual block coefficients vectors $\alpha_{k,l}$ may be concatenated to give a vector α_k of all coefficients for grid k , with corresponding block-diagonal dictionary Φ_k :

$$\alpha_k = \begin{pmatrix} \alpha_{k,0} \\ \alpha_{k,1} \\ \vdots \end{pmatrix} \quad \Phi_k = \begin{pmatrix} \Phi_{k,0} & 0 & \cdots \\ 0 & \Phi_{k,1} & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix}.$$

Operators R_k and B_k are defined in the same way as Φ_k , i.e. as block diagonal matrices with components $R_{k,l}$ and $B_{k,l}$ respectively, and $Q_k^T = (Q_{k,0}^T \ Q_{k,1}^T \ \dots)$. Using these definitions, penalty (1) for grid k may be expressed as

$$\frac{1}{2} \|R_k \Phi_k \alpha_k - R_k B_k \mathbf{s}\|_2^2$$

and penalty (2) for grids k and m may be expressed as

$$\frac{1}{2} \left\| G_m \sum_l Q_{k,l}^T \Phi_{k,l} \alpha_{k,l} - G_k \sum_n Q_{m,n}^T \Phi_{m,n} \alpha_{m,n} \right\|_2^2 = \frac{1}{2} \|G_m Q_k^T \Phi_k \alpha_k - G_k Q_m^T \Phi_m \alpha_m\|_2^2$$

Concatenating the coefficient vectors α_k for each grid gives a single combined coefficient vector α , with corresponding block-diagonal dictionary Φ :

$$\alpha = \begin{pmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \end{pmatrix} \quad \Phi = \begin{pmatrix} \Phi_0 & 0 & \cdots \\ 0 & \Phi_1 & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix}.$$

Making corresponding definitions of R, B , and Q^T (i.e. $R = \text{diag}(R_0, R_1, \dots)$, $B = \text{diag}(B_0, B_1, \dots)$, and $Q^T = \text{diag}(Q_0^T, Q_1^T, \dots)$), and defining

$$G = \begin{pmatrix} -G_0 & G_1 & 0 & \cdots & 0 \\ 0 & -G_1 & G_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & -G_{N_G-2} & G_{N_G-1} \\ G_0 & 0 & \cdots & 0 & -G_{N_G-1} \end{pmatrix},$$

one can express penalty (1) over all grids as $\frac{1}{2} \|R\Phi\alpha - R B \mathbf{s}\|_2^2$, and penalty (2) for all grid pairs 0-1, 1-2, etc. as $\frac{1}{2} \|G Q^T \Phi \alpha\|_2^2$. Penalty (3) is expressed as $\|\alpha\|_1$, the ℓ^1 norm of the combined coefficient vector α .

The resulting functional optimization, combining all of these penalties, may be expressed as

$$\arg \min_{\alpha} \frac{\gamma_0}{2N_0} \|R\Phi\alpha - R B \mathbf{s}\|_2^2 + \frac{\gamma_1}{2N_1} \|G Q^T \Phi \alpha\|_2^2 + \lambda \|\alpha\|_1,$$

where N_0 and N_1 are the lengths of vectors $R\Phi\alpha$ and $G Q^T \Phi \alpha$ respectively, and are included so that the effect of weights γ_0 and γ_1 does not depend on these lengths. Defining

$$A = \begin{pmatrix} \sqrt{\frac{\gamma_0}{N_0}} R\Phi \\ \sqrt{\frac{\gamma_1}{N_1}} G Q^T \Phi \end{pmatrix} \quad \mathbf{b} = \begin{pmatrix} \sqrt{\frac{\gamma_0}{N_0}} R B \mathbf{s} \\ \mathbf{0} \end{pmatrix},$$

this minimization may be written as the standard ℓ^1 problem

$$\arg \min_{\alpha} \frac{1}{2} \|A\alpha - \mathbf{b}\|_2^2 + \lambda \|\alpha\|_1.$$

While a number of very good solvers are available for problems of this structure, many of them do not perform well on this specific problem, and the best results have been obtained using an Iteratively Weighted Least Squares type solver [13].

The final restored image is obtained by averaging all block grid reconstructions $\Phi_k \alpha_k$ to obtain a single value for each image pixel in the inpainting region. The most significant free parameters (typical values in parentheses) of the algorithm are

λ (0.2 to 0.02), γ_0 and γ_1 (both set to 1), block size (between 7×7 and 9×9), number of block grids and offset between them (between 2 and 4 grids, with offset depending on the block size), the size of the dictionary to be associated with each block (the size of a single block vector multiplied by a factor of 4), and the fraction of known block content threshold used to determine how to construct the dictionary (0.75). While the typical values generally give good performance, manual setting of the parameters is required to obtain the best possible results. (Automated parameter optimization is a subject of future research, but it may be argued that some user tunable parameters are desirable given the subjective nature of the inpainting problem.) Performance is moderately insensitive to most of the parameters, but the choice of block size and grid offsets can have a significant effect when there is complex image structure on the inpainting region boundary.

III. RESULTS

The assignment of objective quality measures to inpainted images is a difficult problem [14], but when inpainting is applied in the context of image restoration as opposed to image editing, and the inpainting region is relatively small, it is not unreasonable to compare with a known reference image using distance measures such as SNR and SSIM [15], with the expectation that large differences in these values should indicate perceptible quality differences. For the results presented here, these distance measures were computed *only* on the inpainting region, so that the variation across different methods is not reduced by averaging over the entire test image. Note also that SSIM values for color images were computed on the image intensities only.

Results for a small subimage cropped from the greyscale version of the “Barbara” image are displayed in Fig. 3. This figure compares the proposed approach (computed using publicly released code [16]) with the methods of Criminisi *et al.* [7] (computed using a publicly available implementation [17]), Wexler *et al.* [8] (computed using a publicly available implementation [18]), and Xu and Sun (reference and result images from [11] kindly provided by J. Sun). In this example, the proposed method is clearly visually superior, and also exhibits the best objective quality measures.

Results for the color version of the “Barbara” image, with two different inpainting regions, are displayed in Fig. 4 and 5, and results for one of the Kodak test images (see Fig. 6) are displayed in Fig. 7. In these comparisons the proposed method and that of Wexler *et al.* [8] are clearly superior to that of Criminisi *et al.* [7], and while the quality difference between the two former methods is smaller, the proposed method leads in terms of objective quality measures, and arguably also has a slight perceptual advantage. (Results for the Xu and Sun [11] method are unfortunately not available for these examples.)

IV. CONCLUSIONS

The proposed method is motivated by a conceptually simple exemplar-based image model, and is implemented via global optimization, thereby avoiding some of the problems associated with the greedy approach of progressively filling in

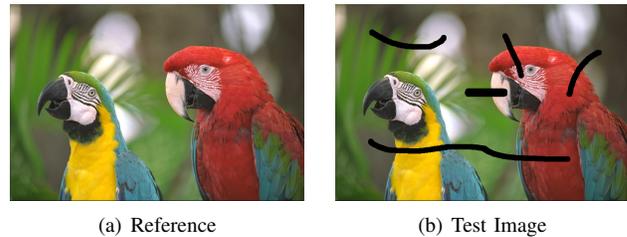


Fig. 6. The “kodim23” image and a derived inpainting test image.

the inpainting region from the region boundary. Results are comparable with, and in some cases better than, other recent inpainting algorithms with which it has been compared. It is computationally expensive, however, and has a large number of free parameters that are difficult to set automatically. It has also been observed that the quality of the results is reduced when dealing with image regions containing irregular texture rather than edges and regular textures (but it should be noted that a similar effect can be observed for other methods that employ sparse linear combinations of exemplars [10], [11]). These issues will be addressed in future research.

REFERENCES

- [1] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, “Image inpainting,” in *Proc. ACM SIGGRAPH*, 2000, pp. 417–424.
- [2] A. Buades, B. Coll, and J.-M. Morel, “Image denoising by non-local averaging,” in *Proc. IEEE ICASSP*, vol. 2, 2005, pp. 25–28.
- [3] K. Dabov, A. Foi, V. Katkovnik, and K. O. Egiazarian, “Image denoising by sparse 3-d transform-domain collaborative filtering,” *IEEE Trans. Image Process.*, vol. 16, no. 8, pp. 2080–2095, 2007.
- [4] J. Yang, J. Wright, T. Huang, and Y. Ma, “Image super-resolution as sparse representation of raw image patches,” in *Proc. IEEE CVPR*, 2008, pp. 1–8.
- [5] D. Glasner, S. Bagon, and M. Irani, “Super-resolution from a single image,” in *Proc. IEEE ICCV*, 2009, pp. 349–356.
- [6] A. A. Efros and T. K. Leung, “Texture synthesis by non-parametric sampling,” in *Proc. IEEE ICCV*, 1999, pp. 1033–1038.
- [7] A. Criminisi, P. Pérez, and K. Toyama, “Region filling and object removal by exemplar-based image inpainting,” *IEEE Trans. Image Process.*, vol. 13, no. 9, pp. 1200–1212, 2004.
- [8] Y. Wexler, E. Shechtman, and M. Irani, “Space-time completion of video,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 3, pp. 463–476, 2007.
- [9] A. Wong and J. Orchard, “A nonlocal-means approach to exemplar-based inpainting,” *Proc. IEEE ICIP*, pp. 2600–2603, 2008.
- [10] B. Shen, W. Hu, Y. Zhang, and Y.-J. Zhang, “Image inpainting via sparse representation,” in *Proc. IEEE ICASSP*, 2009, pp. 697–700.
- [11] Z. Xu and J. Sun, “Image inpainting by patch propagation using patch sparsity,” *IEEE Trans. Image Process.*, vol. 19, no. 5, p. 1, 2010.
- [12] B. Wohlberg, “Inpainting with sparse linear combinations of exemplars,” in *Proc. IEEE ICASSP*, 2009, pp. 689–692.
- [13] P. Rodríguez and B. Wohlberg, “An efficient algorithm for sparse representations with l^p data fidelity term,” in *Proc. 4th IEEE Andean Technical Conf. (ANDESCON)*, 2008.
- [14] P. A. Ardis, C. M. Brown, and A. Singhal, “Inpainting quality assessment,” *J. Electron. Imaging*, vol. 19, no. 1, p. 011002, 2010.
- [15] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: From error visibility to structural similarity,” *IEEE Trans. Image Process.*, vol. 13, no. 4, 2004.
- [16] B. Wohlberg, “Adaptive Algorithms for Image Restoration Matlab Library,” Matlab library: <http://math.lanl.gov/~brendt/Software/LibAAIR/>.
- [17] S. Bhat, “Object removal by exemplar-based inpainting,” Matlab code: <http://www.cc.gatech.edu/~sooraj/inpainting/>, 2004.
- [18] M. Zmiri-Yaniv, M. Ofry, A. Bruker, and T. Hassner, “ImageCompletion,” Executable program: <http://www2.mta.ac.il/~tal/ImageCompletion/>, 2006.

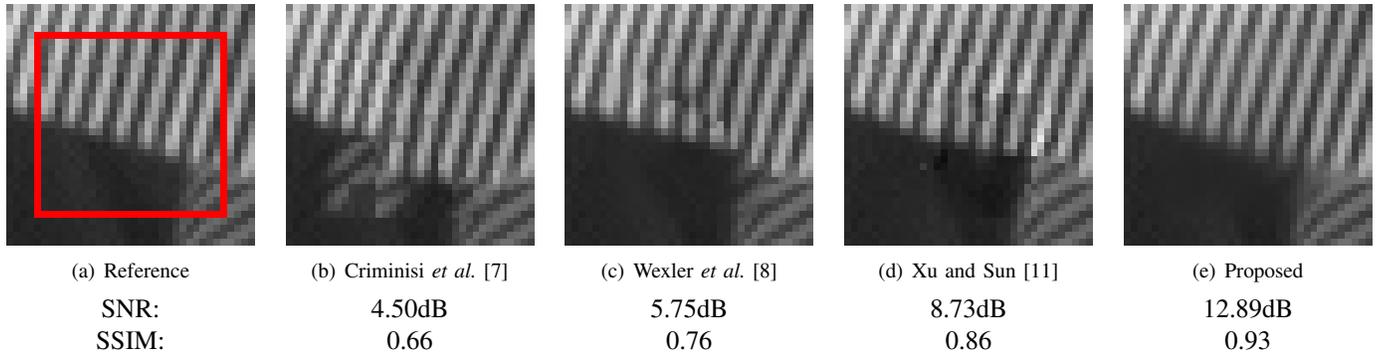


Fig. 3. Inpainting comparison on a subimage cropped from the “Barbara” image (see [11, 3rd row of Fig. 12]). The inpainting region is indicated by a box in the reference image.

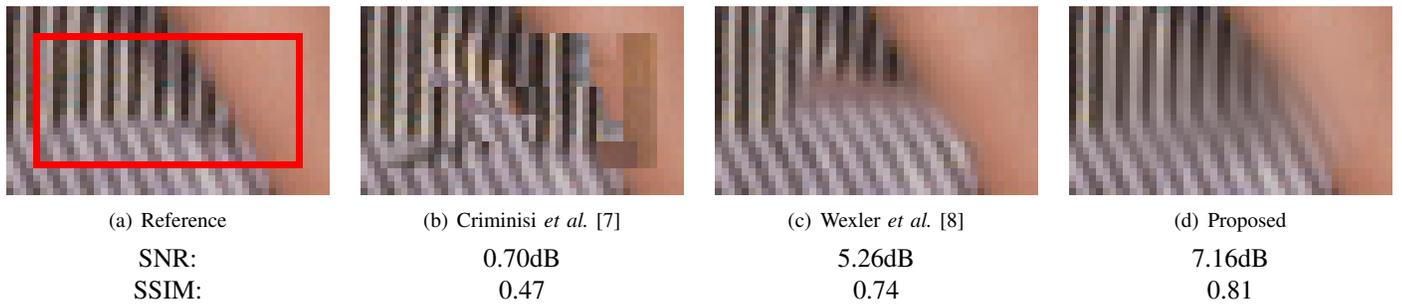


Fig. 4. Comparison of inpainting results for a region of the “Barbara” image. The inpainting region is indicated by a box in the reference image.

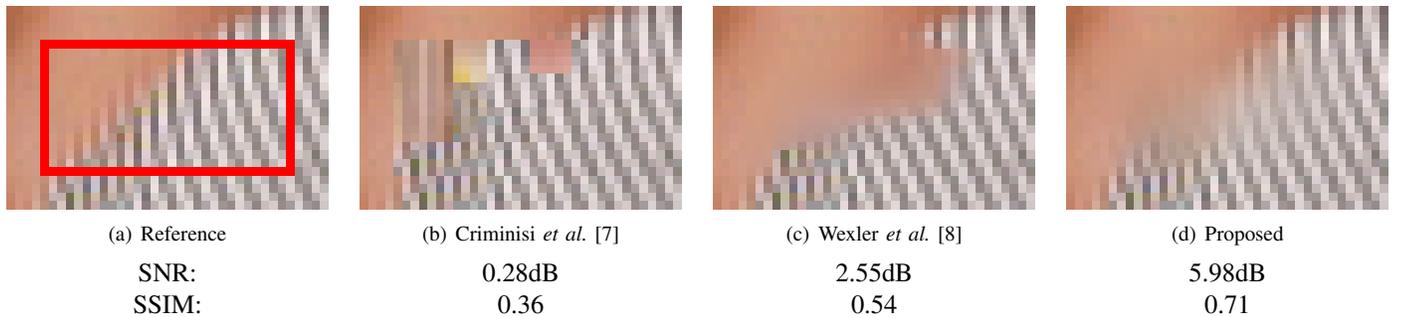


Fig. 5. Comparison of inpainting results for a region of the “Barbara” image. The inpainting region is indicated by a box in the reference image.

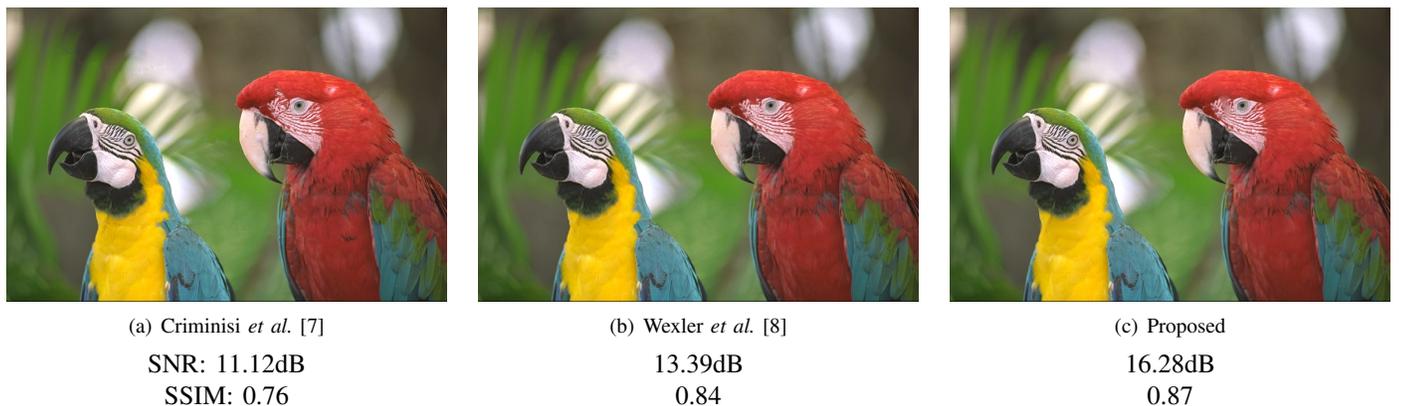


Fig. 7. Comparison of inpainting results for the test image in Fig. 6. (Subtle effects can be observed by zooming in the electronic version of this document, or by viewing the image files included in the supplementary downloadable material available from <http://ieeexplore.ieee.org>.)